PCT ORGANISATION MONDIALE DE LA PROPRIETE INTELLECTUELLE
Bureau international

DEMANDE INTERNATIONALE PUBLIEE EN VERTU DU TRAITE DE COOPERATION EN MATIERE DE BREVETS (PCT)

(51) Classification internationale des brevets 6 :		(11) Numéro de publication internationale:
H04L 9/32	A1	WO 96/33567
		(43) Date de publication internationale: 24 octobre 1996 (24.10.96)

(21) Numéro de la demande internationale: PCT/FR96/00612

(22) Date de dépôt international: 22 avril 1996 (22.04.96)

(30) Données relatives à la priorité:
95/04753 20 avril 1995 (20.04.95) FR
95/07668 27 juin 1995 (27.06.95) FR

(71) Déposant *(pour tous les Etats désignés sauf US)*: GEMPLUS [FR/FR]; Parc d'activités de la Plaine-de-Jouques, Avenue du Pic-de-Bertagne, F-13420 Gemenos (FR).

(72) Inventeur; et
(75) Inventeur/Déposant *(US seulement)*: NACCACHE, David [FR/FR]; 7, rue Chaptal, F-75009 Paris (FR).

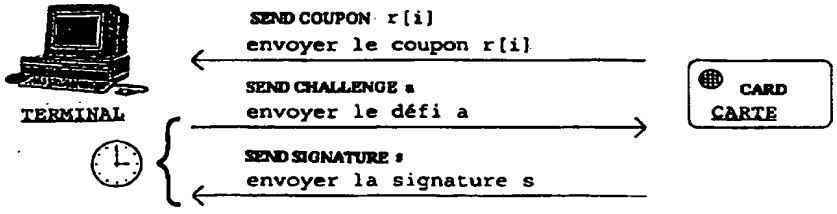
(74) Mandataire: BORIN, Lydie; Cabinet Ballot-Schmit, 16, avenue du Pont-Royal, F-94230 Cachan (FR).

(81) Etats désignés: JP, US, brevet européen (AT, BE, CH, DE, DK, ES, FI, FR, GB, GR, IE, IT, LU, MC, NL, PT, SE).

Publiée
Avec rapport de recherche internationale.

(54) Title: PROCESS FOR GENERATING ELECTRONIC SIGNATURES, IN PARTICULAR FOR SMART CARDS

(54) Titre: PROCEDE DE GENERATION DE SIGNATURES ELECTRONIQUES NOTAMMENT POUR CARTES A PUCES



(57) Abstract

The invention concerns processes for generating digital signatures for electronic messages. The invention proposes modifying signature-generating algorithms, such as DSAs ("Digital Signature Algorithms), in order to enable smart cards with reduced calculation and storage resources to produce digital signatures with a high degree of security in spite of their reduced resources. The signature-checking terminal sends a random number *a* and measures the time taken by the card to send back a signal s using this random number. If the time is greater than a given duration, the signature is rejected even if the check of its authenticity is positive. In addition, part of the signature (the part which does not use the secret card key but only the public algorithm parameters) is precalculated and stored in the card in the form of signature portions produced by a compression function such that they are short. Only the second part of the signature has to be calculated by the card. According to the invention, the calculations to be made are simple so that the card does not require extensive calculation and memory resources.

(57) Abrégé

L'invention concerne les procédés de génération de signature numérique de messages électroniques. L'invention propose de modifier les algorithmes de génération de signature tels que DSA ("Digital Signature Algorithm") pour permettre à des cartes à puces à faibles ressources de calcul et de mémoire de produire des signatures numériques avec un haut degré de sécurité malgré leurs faibles ressources. On prévoit que le terminal de vérification de signature envoie un nombre aléatoire *a* et chronomètre le temps mis par la carte pour renvoyer une signature s utilisant ce nombre aléatoire. Si le temps est supérieur à une durée déterminée, la signature est rejetée même si la vérification de son authenticité est positive. D'autre part, on prévoit qu'une partie de la signature (partie qui n'utilise pas la clé secrète de la carte mais seulement des paramètres publics de l'algorithme) est précalculée et stockée dans la carte sous forme de coupons de signature obtenus par une fonction de compression de sorte qu'ils ont une faible longueur. Seule la deuxième partie de signature est à calculer par la carte, et on s'arrange pour que les calculs à effecturer soient simples pour que la carte n'ait pas besoin de ressources de calcul et de mémoire importantes.

PROCEDE DE GENERATION DE SIGNATURES ELECTRONIQUES, NOTAMMENT POUR CARTES A PUCES

L'invention concerne un procédé de génération de signatures numériques de messages électroniques.

Le procédé s'applique particulièrement à la signature de messages par des appareils portables du type carte à puce à microprocesseur.

Par exemple, il s'agit de signer des messages envoyés par la carte à un terminal de lecture ou à une autorité centrale; ou encore, il s'agit de faire une transaction (chèque électronique) et de signer cette transaction pour qu'elle puisse être authentifiée d'abord par le terminal de lecture dans lequel est faite la transaction, ensuite par une autorité centrale qui gère les transactions.

Le procédé qui va être décrit est apparenté aux algorithmes de génération de signatures numériques qui ont été publiés ces dernières années, notamment par le US National Institute of Standards and Technology, tel que l'algorithme DSA (Digital Signature Algorithm) décrit dans la demande de brevet US 07/738431 et annoncé le 30 Août 1991 au Registre Fédéral tenu par cet Institut, pages 42980-42982.

L'invention a pour but de modifier les procédés connus, notamment pour les rendre adaptables à des cartes à microprocesseur qui n'ont pas des ressources matérielles (processeur, mémoires) suffisantes pour réaliser rapidement des opérations mathématiques sur des grands nombres. Les algorithmes connus, notamment l'algorithme DSA, utilisent des grands nombres pour générer les signatures avec un degré de sécurité suffisant.

Pour mieux faire comprendre l'invention, on va d'abord rappeler ce qu'est l'algorithme DSA.

Une signature DSA est constituée par une paire {r, s} de grands nombres représentés dans les calculateurs par des chaînes longues de chiffres binaires (160 chiffres). La signature numérique est calculée à l'aide d'une série de règles de calcul, définies par l'algorithme, et d'un ensemble de paramètres utilisés dans ces calculs. La signature permet à la fois de certifier l'identité du signataire (parcequ'elle fait intervenir une clé secrète propre au signataire) et l'intégrité du message signé (parcequ'elle fait intervenir le message lui-même). L'algorithme permet d'une part de générer des signatures, et d'autre part de vérifier des signatures.

La génération de signature DSA fait intervenir une clé secrète. La vérification fait intervenir une clé publique qui correspond à la clé secrète mais ne lui est pas identique. Chaque utilisateur possède une paire de clés (secrète, publique). Les clés publiques peuvent être connues de tous, alors que les clés secrètes ne sont jamais dévoilées. Toute personne a la capacité de vérifier la signature d'un utilisateur en utilisant la clé publique de celui-ci, mais seul le possesseur de la clé secrète peut générer une signature correspondant à la paire de clés.

Les paramètres de l'algorithme DSA sont les suivants :

- un nombre premier p tel que $2^{L-1} < p < 2^{L}$ pour L compris entre 512 et 1024 (bornes comprises), et L = 64a pour un a entier quelconque;
- un nombre premier q tel que $2^{159} < q < 2^{160}$ et p-1 est un multiple de q;
- un nombre g, d'ordre q modulo p, tel que :

$g = h^{(p-1)/q}$ modulo p, où h est un entier quelconque vérifiant

$1 < h < p-1$ et $g > 1$;

- un nombre x généré aléatoirement ou pseudo-aléatoirement (c'est la clé secrète, figée pour un utilisateur donné);

- un nombre y défini par la relation

$y = g^x$ modulo p; (c'est la clé publique liée à la clé secrète); les opérations modulaires définies ci-après, modulo p ou modulo q seront désignés par mod p ou mod q respectivement;

- un nombre k généré aléatoirement ou pseudo-aléatoirement, tel que $0 < k < q$.

Les entiers p, q, et g sont des paramètres du système pouvant être publiés et/ou partagés par un groupe d'utilisateurs. Les clés, secrète et publique, d'un signataire sont respectivement x et y. Le paramètre k, aléatoire, doit être régénéré pour chaque nouvelle signature. Les paramètres x et k sont utilisés pour la génération de signatures et doivent être gardés secrets.

Afin de signer un message m (qui sera en général une valeur hachée d'un fichier initial M), le signataire calcule la signature {r, s} par :

$r = (g^k \bmod p) \bmod q$, et

$s = (m + xr)/k \bmod q$

(où la division par k s'entend modulo q, c'est-à-dire que 1/k est le nombre k' tel que $kk' = 1 \bmod q$; par exemple si q=5 et k = 3, alors 1/k = 2 car 3x2 = 6, soit 1 mod 5).

Après avoir testé que r et s sont différents de zéro, la signature {r, s} est envoyée au vérifieur. Le vérifieur est en général le terminal dans lequel est insérée la carte à puce qui envoie le message m et la signature {r, s}.

Le vérifieur, qui connaît p, q, g (liés à l'application), y (lié à l'utilisateur), et m (le message qu'il a reçu de la carte), calcule :

a. $w = (1/s) \bmod q$

b. $u1 = mw \bmod q$

c. $u2 = rw \bmod q$

d. $v = [g^{u1}.y^{u2} \bmod p] \bmod q$

Or cette valeur $[g^{u1}.y^{u2} \bmod p] \bmod q$ est justement égale à r si s a la valeur $(m + xr)/s \bmod q$.

Par conséquent, le terminal reçoit r et s et vérifie que v est bien égal à r pour accepter la signature, ou la rejeter dans le cas contraire.

Dans ce qui suit, on utilisera indifféremment les termes de signataire ou organe signataire, ou dispositif prouveur, ou de carte à puce, pour désigner le dispositif qui émet la signature et qui sera en général une carte à puce. Et on utilisera indifféremment le terme de vérifieur, ou organe vérifieur ou dispositif vérifieur, ou terminal vérifieur, ou encore autorité de contrôle, pour désigner le dispositif qui reçoit la signature et la vérifie pour accepter ou rejeter une transaction ou un message. L'application la plus simple de l'invention est l'émission d'une signature par une carte à puce vers un terminal de lecture dans lequel la carte est insérée, le terminal exécutant la fonction de vérification et étant relié ou non à une autorité centrale de gestion.

Un des buts de la présente invention est d'augmenter la sécurité de génération et vérification de signatures électroniques numériques, en minimisant les moyens de calcul et de mémoire qui doivent être présents dans la carte à puce pour produire les signatures.

Il serait en particulier souhaitable de pouvoir utiliser dans la carte des microprocesseurs peu chers à 8 bits, malgré le fait qu'ils ne peuvent pas facilement

traiter des grands nombres, plutôt que des microprocesseurs plus puissants et plus coûteux. Mais cela ne doit pas se faire au détriment de la sécurité.

Selon un premier aspect important, l'invention propose que la vérification par un vérifieur (terminal) de la signature envoyée par le signataire (carte) utilise une étape de chronométrage de la durée s'écoulant entre un instant où une donnée (en principe aléatoire) est envoyée par le vérifieur au signataire (carte) et l'instant où la signature (utilisant cette donnée aléatoire) revient au vérifieur. Si le temps écoulé est trop long, c'est que le traitement de calcul de signature par le signataire s'effectue de manière anormale et la signature est rejetée même si son authenticité est confirmée par le vérifieur.

Indirectement, cette solution permet, comme on le verra, de conserver la même sécurité de signature tout en utilisant des ressources matérielles faibles (puissance de calcul et mémoires) dans la carte à puce. Des ressources faibles entraînent la nécessité de modifier les procédés de génération et vérification de signatures, mais c'est au détriment de la sécurité. L'étape de chronométrage selon l'invention restaure un niveau de sécurité suffisant.

On décrira en détail cette solution à partir d'algorithmes dérivés de l'algorithme DSA rappelé ci-dessus, mais on comprendra que ce premier aspect de l'invention est applicable avec d'autres algorithmes même s'ils sont très différents de l'algorithme DSA.

En résumé, le premier aspect de l'invention consiste dans un procédé de signature électronique, comportant la génération d'une signature numérique par un organe signataire qui calcule cette signature en utilisant une donnée aléatoire envoyée par un organe vérifieur, et la vérification de la signature par le

vérifieur qui vérifie si une condition mathématique faisant intervenir la signature envoyée et la donnée aléatoire est remplie, ce procédé étant caractérisé en ce que la vérification de la signature envoyée par le signataire au vérifieur utilise en outre une étape de chronométrage de la durée s'écoulant entre un instant où la donnée aléatoire est envoyée par le vérifieur au signataire et l'instant où la signature utilisant cette donnée revient au vérifieur après calcul par le signataire, la signature étant acceptée si le temps écoulé est inférieur à une seuil déterminé et si la condition mathématique est vérifiée.

De préférence, l'algorithme utilisé est du type dans lequel la génération de signature produit deux valeurs {r, s}, s étant calculée à partir de r et d'une clé secrète x, et dans lequel la vérification de la signature {r, s} consiste dans la vérification d'une égalité v = f(r, s) = r entre r et une fonction f de r et de s. On prévoit alors selon l'invention que la fonction f est choisie suffisamment complexe pour que la durée de recherche d'une valeur s à partir de cette égalité en l'absence de connaissance de la clé secrète soit très supérieure, même si elle est faite par un calculateur puissant, à la durée de calcul et transmission par la carte de la valeur s à partir de r et de la clé secrète, et ceci même si la carte utilise un microprocesseur peu puissant (microprocesseur de 8 bits à 20 MHz par exemple). Ainsi, en choisissant correctement la condition de temps introduite par le chronométrage, on fait en sorte que cette condition ne puisse pas être remplie en l'absence de connaissance de la clé secrète et notamment ne puisse pas être remplie par une recherche de s à partir de l'égalité r = f(r, s).

En pratique, la fonction f(r, s) fait intervenir aussi un message m à signer, de sorte qu'on peut la noter f(r, s, m).

De préférence, la fonction f comporte des calculs mathématiques suivis d'une fonction de hachage complexe. La première partie de signature r est établie par d'autres calculs mathématiques, suivis de la même fonction de hachage complexe.

Cette fonction de hachage complexe est de préférence, comme on l'expliquera plus loin, une fonction de compression complexe aboutissant à une réduction de la longueur des chaînes de bits obtenues par les calculs mathématiques effectués.

On rappelle qu'une fonction de hachage est une fonction de traitement logique de chaînes binaires, qui permet d'obtenir une chaîne de caractères de longueur déterminée à partir d'une autre chaîne de caractères de même longueur ou de longueur différente. Une fonction de hachage complexe peut être obtenue par des hachages successifs et/ou des calculs mathématiques impliquant les résultats de plusieurs hachages. Une compression peut être obtenue à la fin en prenant comme résultat une valeur modulaire, modulo 2^e, où e est la longueur de la chaîne finalement désirée.

Par ailleurs, selon un autre aspect important de l'invention, on propose une nouvelle solution pour traiter des plus petits nombres dans la carte à puce, dans des algorithmes de signature numérique du genre dans lequel la signature fait intervenir deux nombres, r et s, seul le nombre s faisant intervenir la clé secrète de la carte et le message à envoyer.

Ce deuxième aspect de l'invention est un perfectionnement à un procédé de génération de signatures qui a été décrit dans la demande de brevet français 93 14466. Dans cette demande de brevet, il est expliqué que

dans un algorithme de ce genre (DSA en est un exemple), le nombre r ne dépend ni du message m envoyé par la carte, ni de la clé secrète contenue dans la carte. Il ne dépend que de nombres figés pour l'application considérée, et de nombres aléatoires; par exemple, ces nombres sont g, p, q et k dans l'algorithme DSA. Il est donc inutile de faire calculer r par la carte, car cela consomme un temps de calcul important. On fait plutôt calculer à l'avance par une autorité centrale certifiée une série de n valeurs r possibles, notées r_i, i étant un indice allant de 1 à n. On stocke les valeurs r_i dans la carte. A chaque nouvelle utilisation de la carte, on utilise une des valeurs r_i (et on n'utilisera plus cette valeur les fois suivantes). Au moment de signer, la carte calcule seulement l'autre partie de signature s, à partir d'une valeur r_i, de la clé secrète x, du message m, et on envoie au vérifieur le message m et le couple $\{ r_i, s\}$ représentant la signature que le vérifieur peut alors vérifier de la manière prévue par l'algorithme considéré.

Les nombres r_i sont des certificats précalculés, appelés encore des "coupons de signature". Ils constituent une partie seulement de la signature à envoyer, et ils peuvent être préparés et stockés à l'avance dans la carte. L'indice i représente l'indice de coupon utilisé lors d'une signature donnée.

Mais une des difficultés réside dans la grande longueur de ces coupons (160 bits dans l'algorithme DSA présenté ci-dessus). Ils consomment une place importante de mémoire non volatile dans la carte; on ne peut pas en sauvegarder un grand nombre dans la carte si on dispose d'une taille limitée de mémoire non volatile; et en plus, ils entraînent un plus long temps de calcul avec un microprocesseur 8 bits puisqu'il faut aller chercher ces nombres par petits morceaux. Mais si on utilisait et stockait des plus petits coupons de signature, la

garantie d'authenticité de signature risquerait d'être bien plus faible.

L'invention décrite ici permet de concilier le souci d'une garantie d'authenticité avec l'utilisation de plus petits coupons de signature r_i.

L'invention propose donc un procédé de génération de signature électronique par un organe signataire et de vérification par un organe vérifieur, utilisant un algorithme de signature numérique dans lequel la signature envoyée par le signataire comprend au moins un coupon de signature r_i et un complément de signature s qui est calculé à partir du coupon r_i et d'une clé secrète x de la carte, cet algorithme permettant la vérification de signature par un vérifieur à l'aide d'une formule de vérification du type

$$v = f(r_i, s) = r_i,$$

ce procédé étant caractérisé en ce que

a. le coupon de signature est établi à l'avance par une autorité certifiée, en deux étapes :

- calcul d'un nombre représenté par une chaîne binaire longue, à l'aide d'une formule mathématique faisant intervenir des grands nombres binaires;

- et modification du résultat de ce calcul par une fonction de compression complexe réduisant fortement la longueur de ce résultat,

b. une série de coupons différents de faible longueur sont ainsi préparés à l'avance et stockés dans l'organe signataire (carte à puce à mémoire et microprocesseur),

c. la génération de signature comporte l'envoi d'un coupon r_i et d'un complément de signature s calculé à partir d'au moins r_i et x,

d. l'algorithme de vérification de signature comporte un calcul mathématique suivi de la même fonction

de compression complexe que celle qui a servi à l'élaboration du coupon, et le résultat est comparé au coupon pour la vérification de signature.

La fonction de compression est de préférence une fonction de hachage complexe qui nécessite un temps de calcul assez long. Ceci donne une sécurité importante au procédé de génération et de vérification de signature. On combine donc l'avantage d'une bonne garantie d'authenticité de signature avec la possibilité de ne sauvegarde dans la carte que des coupons de petite taille, donc la possibilité d'en sauvegarde beaucoup. Si de surcroît on utilise le chronométrage mentionné plus haut, on conçoit qu'on peut renforcer à un très haut degré la garantie d'authenticité.

Le calcul de la signature s fait bien sûr intervenir le message m qu'on veut signer, pour garantir non seulement l'authenticité de la signature mais aussi l'intégrité du message transmis.

On peut encore améliorer la sécurité par une ou plusieurs des caractéristiques suivantes :

La formule de calcul du coupon r_i est de préférence établie à partir d'un aléa J engendré au départ par la carte et stocké dans la carte pour être réutilisé lorsque le coupon sera utilisé pour l'établissement d'une signature.

On peut prévoir que pour déclencher la génération d'une signature, le terminal vérifieur envoie un aléa *a* à la carte et déclenche alors le chronomètre; on prévoit aussi que l'établissement du complément de signature utilise nécessairement cet aléa *a* et que la vérification de signature nécessite également cet aléa *a*.

Le complément de signature s est de préférence établi par un calcul faisant intervenir une fonction de hachage SHA(m, a) du message et de cet aléa *a*, la même

fonction de hachage étant utilisée pour la vérification de signature.

Le complément de signature s est de préférence établi par un calcul faisant intervenir un aléa J stocké dans la carte et ayant servi à établir le coupon de signature. De préférence encore, ce calcul de s fait intervenir une fonction de hachage SHA(x, J, i) portant sur cet aléa J et sur un indice i représentant le numéro du coupon utilisé, cette même fonction de hachage ayant été précédemment utilisée au cours du calcul de chaîne binaire longue prévu dans le calcul du coupon correspondant. Cette fonction de hachage fait de préférence aussi intervenir la clé secrète x de la carte.

Le complément de signature s est de préférence établi par un calcul faisant intervenir une fonction de hachage du coupon $SHA(r_i)$, la même fonction de hachage $SHA(r_i)$ étant utilisée pour la vérification de signature.

Ainsi, selon un aspect particulier de l'invention, on propose un procédé de génération de signatures numériques de messages par un dispositif signataire et de vérification de ces signatures par un dispositif vérifieur, le dispositif signataire comportant des moyens de calcul, de communication et de rétention de données comprenant au moins une mémoire non volatile programmable électriquement, selon lequel on prépare des données chiffrées constituant des coupons de signature r_i que l'on charge dans la mémoire non-volatile et que le dispositif signataire utilise pour signer des messages, principalement caractérisé en ce que :

- les coupons sont compressés par application d'une fonction de compression, dite encore fonction de hachage, par une autorité certifiée avant d'être chargés dans la mémoire, et en ce qu'il comporte les échanges suivants :

- un message m est transmis et ce message doit être certifié par une signature;

- le signataire envoie un coupon r_i au vérifieur,

- le vérifieur envoie un nombre aléatoire *a* au signataire et déclenche un chronomètre,

- le signataire calcule la signature s du message et l'envoie au vérifieur,

- le vérifieur arrête le chronomètre et vérifie que la signature a été obtenue par le secret détenu dans la carte et le coupon r_i reçu; cette vérification est faite en vérifiant l'égalité suivante :

$$v = f(r_i, s, m) = r_i$$

- le vérifieur accepte la signature si la condition de vérification $v = r_i$ est remplie et si le temps chronométré ne dépasse pas une durée prédéterminé impartie.

Pour simplifier, dans toute la suite on parlera surtout de carte pour le signataire ou signataire.

D'autres caractéristiques et avantages de l'invention apparaîtront à la lecture de la description détaillée qui suit et qui est faite en référence aux dessins annexés dans lesquels :

- la figure 1 décrit l'organigramme d'une carte mettant en oeuvre le système proposé par la présente invention;

- la figure 2 décrit les données transmises entre la carte et le terminal au moment de l'utilisation du coupon;

- la figure 3 décrit l'organigramme d'un terminal mettant en oeuvre le système proposé par la présente invention;

- la figure 4 représente les données transmises entre la carte et l'autorité pendant la phase de

chargement des coupons et l'organisation de la mémoire d'une carte après le chargement de n coupons.

A partir des explications données en préambule, on aura compris que le principal avantage des coupons de signature précalculés selon la méthode de l'invention réside dans la vitesse de calcul d'une signature par une carte basée sur un simple microcontrôleur de 8 bits et le faible taux d'occupation de mémoire des coupons stockés. Typiquement le calcul de signature peut se faire en 300ms environ, temps de transmission compris, et chaque coupon peut utiliser de deux à quatre octets de mémoire EPROM ou EEPROM.

On va décrire l'invention dans cet exemple, étant entendu que ce n'est qu'un exemple, bien qu'il soit considéré ici comme le plus avantageux.

Le procédé de génération de signatures se décompose dans ce cas en deux phases distinctes : le chargement des coupons par l'autorité ayant délivré la carte, puis l'utilisation de ces coupons par la carte, face à un terminal ne connaissant pas le secret x de la carte.

Les deux phases font ici appel à des fonctions de hachage de deux types différents. On rappelle qu'une fonction de hachage d'un nombre, représenté par une chaîne de bits, consiste en la production d'une autre chaîne de bits de longueur déterminée, longueur qui est ou non la même que celle de la chaîne de départ, et ceci à partir de fonctions logiques exécutées sur des groupes de bits de la chaîne de départ.

Des fonctions de hachage simples sont utilisées, notées SHA(ch) pour le hachage d'une chaîne ch. Ces fonctions peuvent être des fonctions de hachage classiques, telles que celles publiées dans la récente norme américaine SHA (Secure Hash Algorithm - FIPS PUB

XX, du 1er Février 1993, dans "Digital Signature Standard"). Ces fonctions peuvent être la fonction MDA ou MD5 ou un hachage basé sur l'algorithme DES (Data Encryption Standard).

D'autres fonctions, dites hachage complexe, seront utilisées aussi. Leur caractéristique utilisée ici n'est pas tant d'être une fonction de hachage que d'être une fonction de ralentissement imposée lors de certains traitements de signaux, et aussi d'être une fonction de compression réduisant la longueur des coupons de signature qu'on veut sauvegarde dans la carte à puce.

Cette fonction de ralentissement et de compression est notée ci-après H(ch) pour le traitement d'une chaîne ch.

Toutes sortes de fonctions de ralentissement et compression pourraient être utilisées dans l'invention. A titre d'exemple on a pris comme fonction H(ch) la fonction suivante, où SHA(ch) désigne une fonction de hachage classique :

$$H(ch) = SHA[SHA\{SHA(ch)\}^{SHA(ch)} \bmod p] \bmod 2^e ,$$

où e est la longueur désirée pour les coupons, par exemple 16 à 40 bits soit quelques octets.

Dans tout ce qui suit, on reprendra un algorithme directement inspiré de l'algorithme DSA, pour montrer comment on met en oeuvre les particularités originales de l'invention. Les paramètres p, q, g, x, y utilisés sont ceux définis précédemment à propos de l'algorithme DSA.

CHARGEMENT DE COUPONS DANS LA CARTE

C'est l'étape préliminaire, mais bien sûr seulement dans le cas où on calcule à l'avance, en dehors de la carte, la première partie r de la signature {r, s}

et où on charge plusieurs valeurs possibles r_i dans la carte.

1. La carte remet à zéro un compteur en mémoire non volatile (EPROM ou EEPROM), génère un aléa J (de 10 à 20 octets par exemple), l'enregistre en mémoire non-volatile, et l'envoie à l'autorité de contrôle qui connaît le secret x de la carte et qui calcule, pour i = 1 à n, plusieurs valeurs k_i et plusieurs valeurs r_i :

$$k_i = \{1/(SHA(x, J, i)\} \bmod q$$

et $r_i = H(g^{ki} \bmod p)$; H est la fonction de ralentissement et de compression.

On pourrait envisager aussi que la carte calcule pour chaque i la valeur SHA(x, J, i) et l'envoie à l'autorité de contrôle; celle-ci calcule les nombres r_i.

2. L'autorité envoie les nombres r_i à la carte qui les stocke en mémoire, en conservant le lien avec le repère i. Les nombres k_i ne sont pas conservés.

Si on se réfère à l'algorithme DSA, k_i représente le nombre k aléatoire, modifié à chaque nouvelle signature. Mais au lieu d'être émis par le terminal vérifieur au moment d'une signature, il sera recalculé au moment opportun par la carte. Comme il dépend de i et qu'un coupon d'indice i n'est utilisé qu'une fois, k_i est renouvelé à chaque fois.

UTILISATION D'UN COUPON POUR SIGNER UN MESSAGE

Lorsque la carte désire signer un message, le protocole suivant est utilisé après transmission du message m (de préférence sous forme de fonction hachée du

véritable message, selon une fonction de hachage connue du terminal qui reçoit le message) :

1. La carte

- extrait l'état i du compteur (représentant l'indice courant de la signature qui va être produite),
- extrait de la mémoire non volatile l'aléa J, le secret x, le coupon r_i correspondant à l'indice i;
- calcule I = SHA(x, J, i); cette valeur I n'est autre que l'inverse modulaire de k_i qui a servi au calcul du coupon r_i ;
- calcule A = $xSHA(r_i) \bmod q$
- incrémente i (pour une prochaine signature)
- envoie r_i au terminal vérifieur; cet envoi représente la première partie de la signature.

2. Le terminal génère alors un aléa *a*, pour déclencher la génération de la deuxième partie de signature s; cet envoi constitue en quelque sorte le lancement d'un défi à la carte car le terminal vérifieur déclenche en même temps un chronomètre pour mesurer le temps de réponse de la carte à ce défi.

La signature s que la carte doit envoyer, compte tenu de la formule de vérification $f(r_i, m, s, a) = r_i$ qui est prévue dans le vérifieur est

$$s = [xSHA(r_i) \bmod q + SHA(m, a)]/ k_i \bmod q$$

Cette formule fait intervenir le coupon r_i, le secret x de la carte, le message m envoyé, le nombre k_i, et l'aléa *a* envoyé par le vérifieur à titre de défi. Cette formule est différente de celle qui a été donnée pour l'algorithme DSA : $s = (m + xr)/k$ pour plusieurs

raisons : elle doit faire intervenir l'aléa a envoyé à titre de défi, pour que le vérifieur soit sûr que le calcul chronométré de signature s ne commence que lorsque l'aléa a est parvenu à la carte. C'est pour cela qu'on utilise un hachage de m et de l'aléa a, SHA (m, a), au lieu de m. D'autre part on utilise de préférence $SHA(r_i)$ plutôt que r_i pour utiliser une valeur de coupon sous forme de chaîne plus longue que r_i qui est une chaîne très courte. Ceci renforce la sécurité. Mais bien entendu, si on utilise $xSHA(r_i)$ au lieu de xr_i et SHA(m, a) au lieu de m, la formule de vérification doit en tenir compte, et on verra plus loin que c'est ce qui est fait. D'autres variantes de calcul de signature peuvent être prévues, à condition simplement que la formule de vérification en tienne compte.

3. La carte calcule, aussi vite que possible, la signature s. Mais comme elle a déjà calculé, avant déclenchement du chronomètre, $A = xSHA(r_i) \bmod q$ et $I = 1/k_i = SHA(x, J, i)$ il ne lui reste qu'à calculer

$$s = I.(SHA(m, a) + A) \bmod q$$

Ce calcul peut être rapide même pour un microcontrôleur simple et peu coûteux de 8 bits, par exemple type 8051 de Intel ou 6805 de Motorola. Dès que le calcul est terminé, la carte renvoie la signature s.

4. Dès réception de s, le terminal arrête le chronomètre et effectue les calculs de vérification de l'authenticité de la signature. Si la signature a été correctement calculée selon la formule ci-dessus, alors on peut vérifier qu'on doit avoir l'égalité suivante :

$$[y^{(SHA(ri)/s) \bmod q} \; g^{(SHA(m, a)/s) \bmod q} \bmod p] = g^{ki} \bmod p$$

Le vérifieur ne possède pas k_i. Il possède $r_i = H(g^{ki} \bmod p)$; H est la fonction de ralentissement et de compression.

L'égalité doit donc être transformée en :

$$H[y^{(SHA(ri)/s) \bmod q} \; g^{(SHA(m,a)/s) \bmod q \bmod p}] = H(g^{ki} \bmod p) = r_i$$

Le vérifieur dispose de r_i, de s, de q, de p, de g, de m, de a, de la fonction de hachage simple SHA, et de la fonction de ralentissement et de compression H. Il vérifie donc l'égalité ci-dessus.

Si l'égalité est obtenue et si la signature a été renvoyée dans un délai inférieur à un seuil déterminé, la signature est acceptée par le vérifieur. Si une des deux conditions n'est pas remplie, elle n'est pas acceptée.

A titre d'exemple pour l'évaluation de la durée on peut donner les indications suivantes : appelons T le temps nécessaire pour évaluer H(ch) sur un ordinateur extrêmement puissant, voire le plus puissant qu'on connaisse aujourd'hui. On peut considérer que la fonction de ralentissement H, aboutissant à des chaînes de longueur e (H ayant également une fonctionnalité de compression) est suffisamment complexe, et en tous cas doit être choisie suffisamment complexe, pour que pour toute valeur z et tout ordinateur existant, la recherche d'une nouvelle valeur ch' telle que z = H(ch') nécessite un temps $T.2^e$.

Etant donné que quelqu'un qui ignore le secret de la carte ne peut rechercher s que par tâtonnements à partir de la formule de vérification (recherche exhaustive), il ne pourra pas, même avec un seul essai, trouver une valeur correcte de s si on choisit de mettre un seuil de durée de renvoi de signature très inférieur à

cette valeur $T.2^e$, par exemple 1 millionième de cette valeur.

Ceci donne une indication de la méthodologie à suivre pour choisir la fonction de ralentissement H et la durée de seuil.

De façon générale, les principes qui ont été expliqués ci-dessus et illustrés par un exemple sont applicables à d'autres protocoles de signature. En particulier ils sont applicables à d'autres protocoles dans lesquels un précalcul de coupons de signature est possible, en particulier les protocoles suivants :

- Rueppel-Nyberg : "New signature schemes based on the discrete logarithm problem" publié dans les actes du colloque Eurocrypt 94.
- Schnorr : "Efficient identification and signatures for smart-cards", publié dans les actes du colloque Crypto'89.
- El-Gamal : "A public-key cryptosystem and a signature scheme based on discrete logarithms" publié dans la revue IEEE Transactions on Information Theory, vol IT30, n°4, pages 469-472.
- Guillou-Quisquater : "A practical zero-knowledge protocol fitted to security microprocessors minimizing both transmission and memory", publié dans les actes du colloque Eurocrypt'88 et "A paradoxical identity-based signature scheme resulting from zero-knowledge", publié dans les actes du colloque Crypto'88.
- d'autres systèmes à clé publique basés sur le logarithme discret, où l'équation $(m + xr)/k \bmod q$ est remplacée par une autre égalité faisant intervenir m, x, r, et k (comme expliqué dans l'article "Meta Message Recovery and Meta Blind Signature schemes based on the discrete logarithm problem and their applications", publié par Horster et al. dans les actes du colloque

Asiacrypt'94) ou encore en utilisant plusieurs aléas distincts k ou plusieurs secrets distincts x dans la même signature.

L'invention est applicable à la signature de chèques électroniques et permet alors de faire de tels chèques avec des cartes à puces à faible coût (résultant de l'utilisation d'un microprocesseur de 8 bits et d'une mémoire non volatile de taille limitée).

En effet, le message m peut représenter une transaction effectuée par la carte avec le terminal qui est par exemple le terminal de paiement d'un commerçant. Ce message m est signé. Le terminal vérifie la signature pour accepter le message, donc la transaction, mais ce terminal est également relié à une autorité centrale de gestion (une banque par exemple) qui doit pouvoir elle-même contrôler le message et l'authenticité de la signature avant de débiter le compte du signataire d'une part et/ou créditer le compte du commerçant d'autre part.

Ainsi, après avoir exécuté toute la procédure de signature et vérification de signature décrite en détail ci-dessus, le terminal envoie à l'autorité de contrôle le chèque électronique $\{i, r_i, a, s, m\}$, et l'autorité s'assure que la signature s est la bonne signature, c'est-à-dire que :

$$s = SHA(x, J, i)[SHA(m, a) + xSHA(r_i)] \bmod q$$

et l'autorité crédite le compte du terminal du montant de la transaction définie dans le message m.

On notera que dans le calcul de la signature par la carte, on peut utiliser l'expression SHA(m, i, a) au lieu de SHA(m, a). Auquel cas il faut bien sûr que la formule de vérification par le terminal en tienne compte et soit donc :

$$H[y^{(SHA(ri)/s) \bmod q}\, g^{(SHA(m, i, a)/s) \bmod q} \bmod p] = r_i$$

et que la formule de vérification de signature par l'autorité en tienne compte également et soit :

$$s = SHA(x, J, i)[SHA(m, i, a) + xSHA(r_i)] \bmod q$$

Si on se réfère aux figures, chaque carte à puce se compose d'une unité de traitement (CPU) 11, d'une interface de communication 10, une mémoire vive 13 (RAM) et/ou une mémoire non inscriptible (ROM) 14 et/ou une mémoire non volatile inscriptible ou réinscriptible (EPROM ou EEPROM) 15.

L'unité de traitement 11 et/ou la ROM 14 de la carte à puce contiennent des programmes ou des ressources de calcul correspondant à l'exécution des étapes de calcul effectuées par la carte lors du chargement des coupons et lors de la signature d'un message ou l'émission d'un chèque électronique. Ces programmes comportent notamment les règles de calcul pour la génération de s et les règles d'utilisation de la fonction de hachage SHA. L'unité de calcul et les programmes en ROM comportent également les ressources nécessaires à des multiplications, additions et réductions modulaires. Certaines de ces opérations peuvent être regroupées (par exemple, la réduction modulaire peut être directement intégrée dans la multiplication).

De même que pour l'algorithme DSA, la RAM de la carte contient le message M et l'aléa *a* sur lesquels s'applique la fonction de hachage SHA(m, a) ou SHA(m, i, a) par exemple. La mémoire non volatile 15 contient typiquement les paramètres q, x, J et le jeu de coupons (r_i) précalculés. L'indice i est dans un compteur non volatile incrémenté à chaque nouvelle génération de signature et remis à zéro lors du chargement de coupons.

L'unité de traitement de la carte commande, via des bus d'adresses et de données 16 et l'interface de

communication 10, les opérations de lecture et d'écriture en mémoire 13, 14, et 15.

Chaque carte à puce est protégée du monde extérieur par des protections physiques 17. Ces protections devraient être suffisantes pour empêcher toute entité non autorisée d'obtenir la clé secrète x. Les techniques les plus utilisées de nos jours en la matière sont l'intégration de la puce dans un module de sécurité et l'équipement des puces de dispositifs capables de détecter des variations de température, de lumière, ainsi que des tensions et fréquences d'horloge anormales. Des techniques de conception particulières telles que l'embrouillage de l'accès mémoire sont également utilisées.

Le terminal se compose quant à lui au minimum d'une unité de traitement (CPU) 30 et des ressources mémoires 32, 33, 34.

Le CPU 30 commande, via les bus d'adresse et de données 35 et l'interface de communication 31, les opérations de lecture et d'écriture dans les mémoires 32, 33, 34.

Le CPU 30 et/ou la ROM 34 de l'autorité contiennent des programmes ou ressources de calcul permettant de mettre en oeuvre les règles de calcul et fonctions de hachage, ralentissement et compression, multiplication, addition, inversion modulaire, exponentiation et réduction modulaire, nécessaires au calcul des coupons et à la vérification de signature. Certaines de ces opérations peuvent être regroupées (multiplication et réduction modulaire par exemple).

L'ensemble de l'invention a été décrite a propos de cartes à puces, mais on comprendra qu'elle est applicable lorsque l'organe signataire est un autre objet, et en particulier un objet portable tel que des cartes PCMCIA qui sont des sortes de cartes à puce à

protocoles de transmission parallèle et non série, ou des badges, des cartes sans contacts, etc. La communication peut s'effectuer entre la carte et le terminal soit directement par des signaux électroniques, soit par transmission à distance, hertzienne ou infrarouge.

REVENDICATIONS

1. Procédé de signature électronique, comportant la génération d'une signature numérique (s) par un organe signataire qui calcule cette signature en utilisant une donnée aléatoire (a) envoyée par un organe vérifieur, et la vérification de la signature par le vérifieur qui vérifie si une condition mathématique faisant intervenir la signature envoyée et la donnée aléatoire est remplie, caractérisé en ce que la vérification de la signature envoyée par le signataire au vérifieur utilise en outre une étape de chronométrage de la durée s'écoulant entre un instant où la donnée aléatoire est envoyée par le vérifieur au signataire et l'instant où la signature utilisant cette donnée revient au vérifieur après calcul par l'organe signataire, la signature étant acceptée si le temps écoulé est inférieur à une seuil déterminé et si la condition mathématique est vérifiée.

2. Procédé selon la revendication 1, caractérisé en ce que le calcul de la signature et la vérification sont effectués à partir d'un algorithme du type dans lequel la génération de signature produit deux valeurs {r, s}, s étant calculée par le signataire à partir de r et d'une clé secrète x, et dans lequel la vérification de la signature {r, s} consiste dans la vérification d'une égalité v = f(r, s) = r entre r et une fonction f de r et de s, et en ce que la fonction f est choisie suffisamment complexe pour que la durée de recherche d'une valeur s à partir de cette égalité en l'absence de connaissance de la clé secrète x soit très supérieure, même si elle est faite par un calculateur puissant, à la durée de calcul et de transmission par la carte de la valeur s à partir

de r et de la clé secrète, et ceci même si la carte utilise un microprocesseur peu puissant.

3. Procédé selon la revendication 2, caractérisé en ce que la fonction f(r, s) fait intervenir aussi un message m à signer.

4. Procédé selon l'une des revendications 2 et 3, caractérisé en ce que la fonction f comporte des calculs mathématiques suivis d'une fonction de hachage complexe (H) réalisant à la fois un ralentissement de l'obtention d'un résultat de calcul et une compression de longueur de ce résultat.

5. Procédé selon la revendication 4, caractérisé en ce que la première partie de signature r est établie par d'autres calculs mathématiques, suivis de la même fonction de hachage complexe (H).

6. Procédé de génération de signature et de vérification selon l'une des revendications 1 à 5, caractérisé en ce que la signature envoyée par le signataire comporte au moins un coupon de signature r_i et un complément de signature s qui est calculé à partir du coupon r_i et d'une clé secrète x de la carte, le procédé permettant la vérification de signature par le vérifieur à l'aide d'une formule de vérification du type

$$v = f(r_i, s) = r_i,$$

ce procédé étant caractérisé en ce que

a. le coupon de signature est établi à l'avance par une autorité certifiée, en deux étapes :

- calcul d'un nombre représenté par une chaîne binaire longue, à l'aide d'une formule mathématique faisant intervenir des grands nombres binaires;

- et modification du résultat par une fonction de compression complexe réduisant fortement la longueur de ce résultat,

b. une série de coupons différents de faible longueur sont ainsi préparés à l'avance et stockés dans l'organe signataire,

c. la génération de signature comporte l'envoi d'un coupon r_i et d'un complément de signature s calculé à partir de r_i, et x,

d. la vérification de signature comporte un calcul mathématique suivi de la même fonction de compression complexe que celle qui a servi à l'élaboration du coupon, et le résultat est comparé au coupon pour la vérification de signature.

7. Procédé de génération de signature électronique pouvant utiliser une étape de chronométrage selon la revendication 1, ce procédé comportant la génération d'une signature par un organe signataire et la vérification de la signature par un organe vérifieur, caractérisé en ce que la signature envoyée par le signataire comprend au moins un coupon de signature r_i et un complément de signature s qui est calculé à partir du coupon r_i et d'une clé secrète x de la carte, la vérification de signature par le vérifieur étant effectuée à l'aide d'une formule de vérification du type $v = f(r_i, s) = r_i$, et en ce que :

a. le coupon de signature est établi à l'avance par une autorité certifiée, en deux étapes :

- calcul d'un nombre représenté par une chaîne binaire longue, à l'aide d'une formule mathématique faisant intervenir des grands nombres binaires;

- et modification du résultat par une fonction de compression complexe réduisant fortement la longueur de ce résultat,

b. une série de coupons différents de faible longueur sont ainsi préparés à l'avance et stockés dans l'organe signataire,

c. la génération de signature comporte l'envoi d'un coupon r_i et d'un complément de signature s calculé à partir d'au moins r_i et x,

d. la vérification de signature comporte un calcul mathématique suivi de la même fonction de compression complexe que celle qui a servi à l'élaboration du coupon, et le résultat est comparé au coupon pour la vérification de signature.

8. Procédé selon la revendication 7, caractérisé en ce que la fonction de compression est une fonction de hachage complexe.

9. Procédé selon l'une des revendications 7 et 8, caractérisé en ce que le calcul du coupon est effectué à partir d'un aléa (J) engendré au départ par la carte et stocké dans la carte pour être réutilisé lorsque le coupon sera utilisé pour l'établissement d'une signature.

10. Procédé selon l'une des revendications 7 à 9, caractérisé en ce que, pour déclencher la génération de signature par la carte, l'organe vérifieur envoie un aléa *a* à la carte, déclenche alors un chronomètre, mesure le temps mis par la carte pour renvoyer le complément de signature s calculé à partir d'au moins l'aléa *a* et la clé secrète x de la carte, effectue un calcul de vérification de signature à partir d'au moins la signature s et l'aléa *a*, et accepte la signature si le calcul vérifie une condition prédéterminée et si le temps

mis par la carte pour renvoyer la signature s utilisant l'aléa *a* est inférieur à un seuil prédéterminé.

11. Procédé selon l'une des revendications 7 à 10, caractérisé en ce que le complément de signature s est établi à partir d'une fonction de hachage SHA(m, a) d'un message m à signer et de l'aléa *a*, et en ce que la même fonction de hachage est utilisée pour la vérification de signature.

12. Procédé selon l'une des revendications 7 à 11, caractérisé en ce que le complément de signature est établi par un calcul faisant intervenir un aléa (J) stocké dans la carte et ayant servi à établir le coupon de signature.

13. Procédé selon la revendication 12, caractérisé en ce que ce calcul faisant intervenir l'aléa (J) stocké dans la carte fait aussi intervenir une fonction de hachage SHA(x, J, i) portant au moins sur cet aléa (J) et sur un indice i représentant un numéro du coupon utilisé, cette même fonction de hachage SHA(x, J, i) ayant été précédemment utilisée au cours du calcul de chaîne binaire longue prévu dans le calcul du coupon correspondant.

14. Procédé selon l'une des revendications 7 à 13, caractérisé en ce que le complément de signature est établi par un calcul faisant intervenir une fonction de hachage du coupon, la même fonction de hachage du coupon étant utilisée pour la vérification de signature.

15. Procédé de génération de signatures numériques de messages par un dispositif signataire et de vérification de ces signatures par un dispositif

vérifieur, le dispositif signataire comportant des moyens de calcul, de communication et de rétention de données comprenant au moins une mémoire non volatile programmable électriquement, procédé selon lequel on prépare des données chiffrées constituant des coupons de signature r_i que l'on charge dans la mémoire non-volatile et que le dispositif signataire utilise pour signer des messages, principalement caractérisé en ce que :

- les coupons sont compressés par application d'une fonction de compression (H), dite encore fonction de hachage, par une autorité certifiée avant d'être chargés dans la mémoire,

et en ce qu'il comporte les échanges suivants :

- un message m est transmis et ce message doit être certifié par une signature;
- le signataire envoie un coupon r_i au vérifieur,
- le vérifieur envoie un nombre aléatoire a au signataire et déclenche un chronomètre,
- le signataire calcule la signature s du message et l'envoie au vérifieur,
- le vérifieur arrête le chronomètre et vérifie que la signature a été obtenue par le secret détenu dans la carte et le coupon r_i reçu; cette vérification est faite en vérifiant l'égalité suivante : $v = f(r_i, s, m) = r_i$
- le vérifieur accepte la signature si la condition de vérification $v = r_i$ est remplie et si le temps chronométré ne dépasse pas une durée prédéterminé impartie.

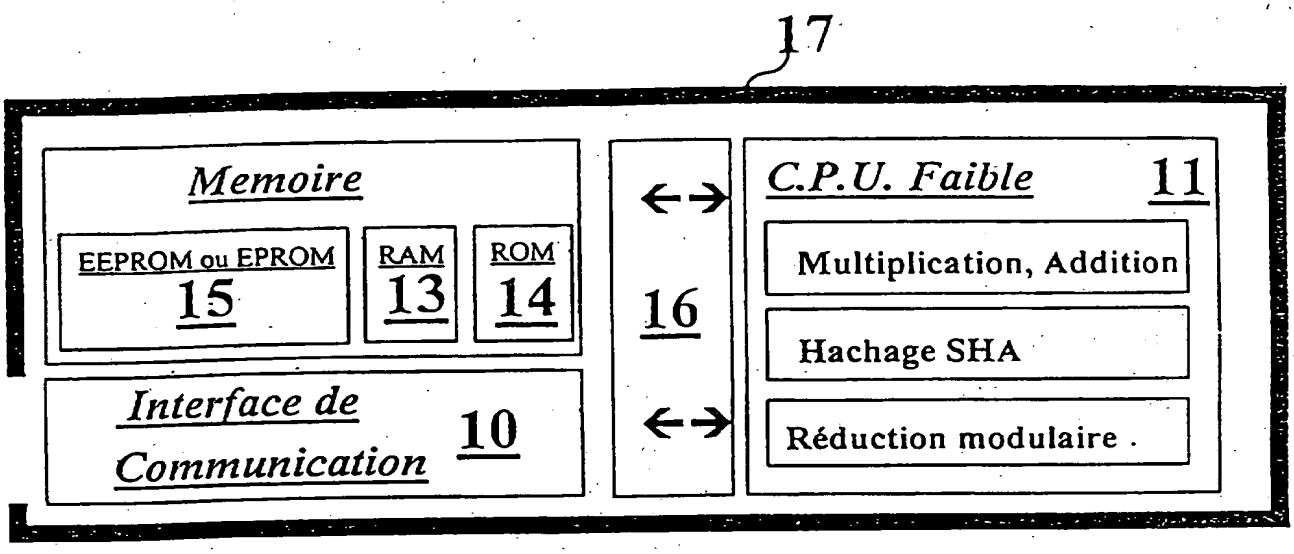

17
Memoire
EEPROM ou EPROM
15
RAM
13
ROM
14
Interface de Communication
10
16
C.P.U. Faible
11
Multiplication, Addition
Hachage SHA
Réduction modulaire .

FIG 1

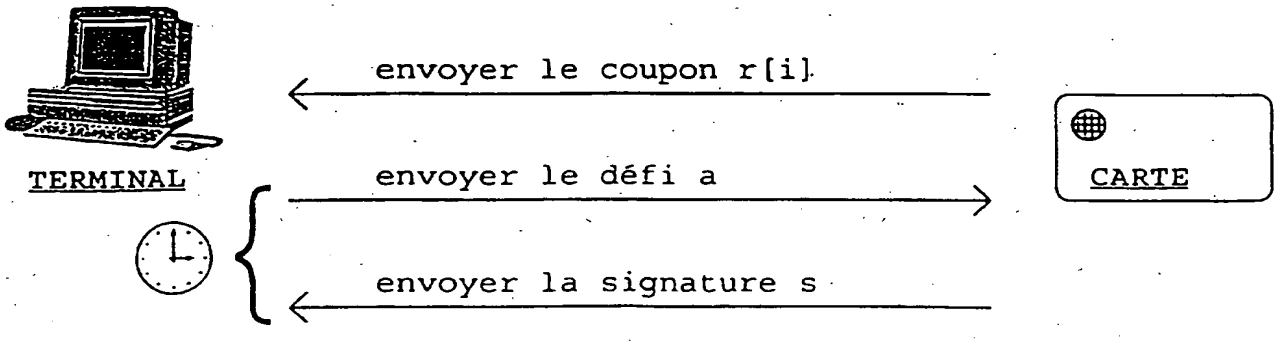

envoyer le coupon r[i]
TERMINAL
envoyer le défi a
CARTE
envoyer la signature s

FIG 2

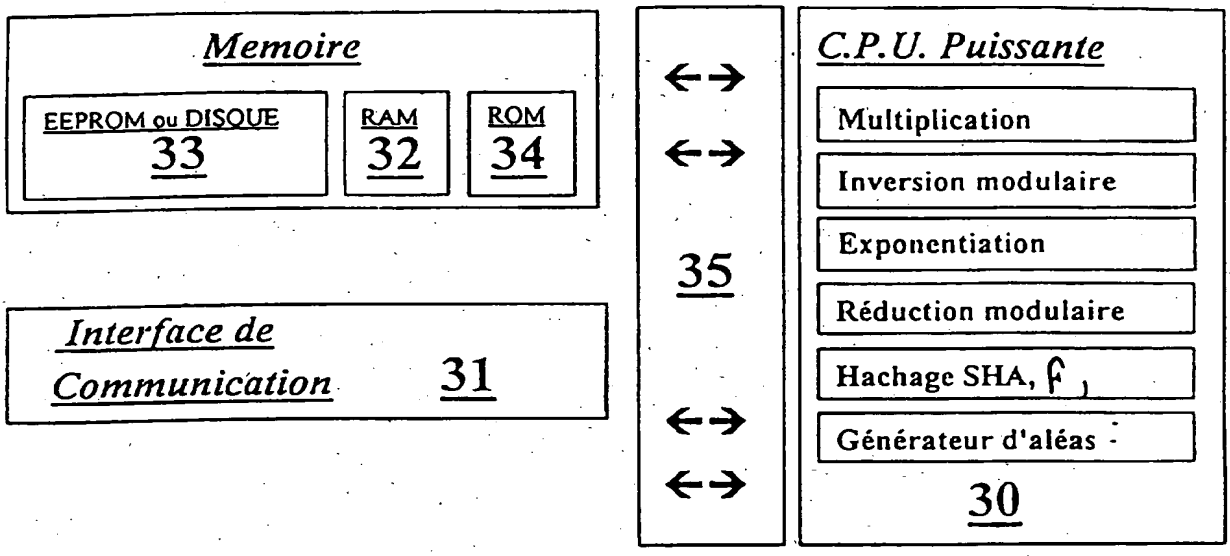

Memoire
EEPROM ou DISQUE
33
RAM
32
ROM
34
Interface de Communication
31
35
C.P.U. Puissante
Multiplication
Inversion modulaire
Exponentiation
Réduction modulaire
Hachage SHA, f
Générateur d'aléas
30

FIG 3

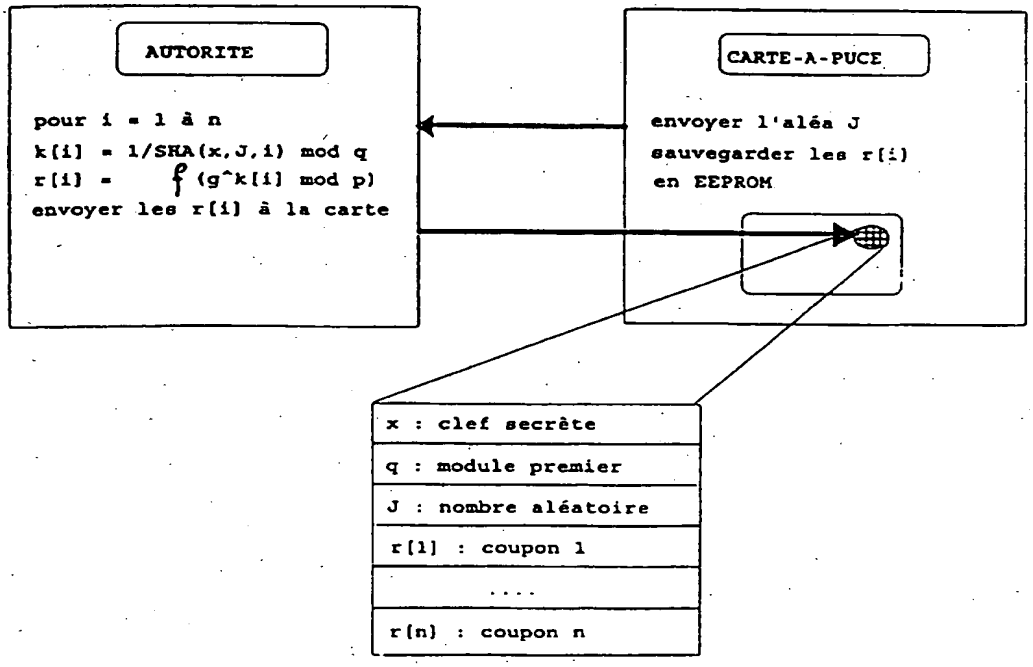

AUTORITE
pour i = 1 à n
k[i] = 1/SHA(x,J,i) mod q
r[i] = f(g^k[i] mod p)
envoyer les r[i] à la carte
CARTE-A-PUCE
envoyer l'aléa J
sauvegarder les r[i]
en EEPROM
x : clef secrète
q : module premier
J : nombre aléatoire
r[1] : coupon 1
....
r[n] : coupon n

FIG 4

INTERNATIONAL SEARCH REPORT

Intern ul Application No
PCT/FR 96/00612

A. CLASSIFICATION OF SUBJECT MATTER
IPC 6 H04L9/32

According to International Patent Classification (IPC) or to both national classification and IPC

B. FIELDS SEARCHED

Minimum documentation searched (classification system followed by classification symbols)
IPC 6 H04L G07F

Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched

Electronic data base consulted during the international search (name of data base and, where practical, search terms used)

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category *	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
X A	EP,A,0 186 038 (CASIO) 2 July 1986 see page 2, line 9 - line 25 see page 3, line 7 - line 25 see page 4, line 4 - line 9 see page 8, line 21 - page 9, line 28 see page 12, line 9 - page 14, line 29 see page 15, line 36 - page 16, line 5 ---	1 15
A	BYTE, vol. 18, no. 12, November 1993, PETERBOROUGH (US), XP000408886 B. SCHNEIER: "DIGITAL SIGNATURES" see page 310, right-hand column, line 15 - page 311, left-hand column, line 11 see page 311, right-hand column, line 14 - line 50 --- -/--	2-4,7,15

[X] Further documents are listed in the continuation of box C.

[X] Patent family members are listed in annex.

* Special categories of cited documents :

"A" document defining the general state of the art which is not considered to be of particular relevance

"E" earlier document but published on or after the international filing date

"L" document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified)

"O" document referring to an oral disclosure, use, exhibition or other means

"P" document published prior to the international filing date but later than the priority date claimed

"T" later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention

"X" document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone

"Y" document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art.

"&" document member of the same patent family

2

Date of the actual completion of the international search	Date of mailing of the international search report
16 July 1996	30. 07. 96

Name and mailing address of the ISA	Authorized officer
European Patent Office, P.B. 5818 Patentlaan 2 NL - 2280 HV Rijswijk Tel. (+31-70) 340-2040, Tx. 31 651 epo nl, Fax: (+31-70) 340-3016	Holper, G

Form PCT/ISA/210 (second sheet) (July 1992)

INTERNATIONAL SEARCH REPORT

Intern al Application No
PCT/FR 96/00612

C.(Continuation) DOCUMENTS CONSIDERED TO BE RELEVANT

Category *	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
A	JOURNAL OF CRYPTOLOGY, 1991, USA, vol. 4, no. 3, ISSN 0933-2790, pages 161-174, XP000573164 SCHNORR C P: "Efficient signature generation by smart cards" see page 172, line 6 - last line -----	2-4,7,15

2

INTERNATIONAL SEARCH REPORT

...ormation on patent family members

Intern... al Application No
PCT/FR 96/00612

Patent document cited in search report	Publication date	Patent family member(s)		Publication date
EP-A-186038	02-07-86	JP-C-	1825638	28-02-94
		JP-B-	5033416	19-05-93
		JP-A-	61139873	27-06-86
		CA-A-	1245764	29-11-88
		FR-A-	2574963	20-06-86
		US-A-	4710613	01-12-87

RAPPORT DE RECHERCHE INTERNATIONALE

Dem. Internationale No
PCT/FR 96/00612

A. CLASSEMENT DE L'OBJET DE LA DEMANDE
CIB 6 H04L9/32

Selon la classification internationale des brevets (CIB) ou à la fois selon la classification nationale et la CIB

B. DOMAINES SUR LESQUELS LA RECHERCHE A PORTE

Documentation minimale consultée (système de classification suivi des symboles de classement)
CIB 6 H04L G07F

Documentation consultée autre que la documentation minimale dans la mesure où ces documents relèvent des domaines sur lesquels a porté la recherche

Base de données électronique consultée au cours de la recherche internationale (nom de la base de données, et si cela est réalisable, termes de recherche utilisés)

C. DOCUMENTS CONSIDERES COMME PERTINENTS

Catégorie *	Identification des documents cités, avec, le cas échéant, l'indication des passages pertinents	no. des revendications visées
X	EP,A,0 186 038 (CASIO) 2 Juillet 1986	1
A	voir page 2, ligne 9 - ligne 25 voir page 3, ligne 7 - ligne 25 voir page 4, ligne 4 - ligne 9 voir page 8, ligne 21 - page 9, ligne 28 voir page 12, ligne 9 - page 14, ligne 29 voir page 15, ligne 36 - page 16, ligne 5	15

A	BYTE, vol. 18, no. 12, Novembre 1993, PETERBOROUGH (US), XP000408886 B. SCHNEIER: "DIGITAL SIGNATURES" voir page 310, colonne de droite, ligne 15 - page 311, colonne de gauche, ligne 11 voir page 311, colonne de droite, ligne 14 - ligne 50	2-4,7,15

	-/--	

[X] Voir la suite du cadre C pour la fin de la liste des documents

[X] Les documents de familles de brevets sont indiqués en annexe

* Catégories spéciales de documents cités:

"A" document définissant l'état général de la technique, non considéré comme particulièrement pertinent

"E" document antérieur, mais publié à la date de dépôt international ou après cette date

"L" document pouvant jeter un doute sur une revendication de priorité ou cité pour déterminer la date de publication d'une autre citation ou pour une raison spéciale (telle qu'indiquée)

"O" document se référant à une divulgation orale, à un usage, à une exposition ou tous autres moyens

"P" document publié avant la date de dépôt international, mais postérieurement à la date de priorité revendiquée

"T" document ultérieur publié après la date de dépôt international ou la date de priorité et n'appartenenant pas à l'état de la technique pertinent, mais cité pour comprendre le principe ou la théorie constituant la base de l'invention

"X" document particulièrement pertinent; l'invention revendiquée ne peut être considérée comme nouvelle ou comme impliquant une activité inventive par rapport au document considéré isolément

"Y" document particulièrement pertinent; l'invention revendiquée ne peut être considérée comme impliquant une activité inventive lorsque le document est associé à un ou plusieurs autres documents de même nature, cette combinaison étant évidente pour une personne du métier

"&" document qui fait partie de la même famille de brevets

Date à laquelle la recherche internationale a été effectivement achevée	Date d'expédition du présent rapport de recherche internationale
16 Juillet 1996	30.07.96

Nom et adresse postale de l'administration chargée de la recherche internationale	Fonctionnaire autorisé
Office Européen des Brevets, P.B. 5818 Patentlaan 2 NL - 2280 HV Rijswijk Tel. (+31-70) 340-2040, Tx. 31 651 epo nl, Fax (+31-70) 340-3016	Holper, G

Formulaire PCT/ISA/210 (deuxième feuille) (juillet 1992)

RAPPORT DE RECHERCHE INTERNATIONALE

Demande internationale No
PCT/FR 96/00612

C.(suite) DOCUMENTS CONSIDERES COMME PERTINENTS

Catégorie *	Identification des documents cités, avec, le cas échéant, l'indication des passages pertinents	no. des revendications visées
A	JOURNAL OF CRYPTOLOGY, 1991, USA, vol. 4, no. 3, ISSN 0933-2790, pages 161-174, XP000573164 SCHNORR C P: "Efficient signature generation by smart cards" voir page 172, ligne 6 - dernière ligne	2-4,7,15

2

RAPPORT DE RECHERCHE INTERNATIONALE

Renseignements relatifs aux membres de familles de brevets

Demande internationale No
PCT/FR 96/00612

Document brevet cité au rapport de recherche	Date de publication	Membre(s) de la famille de brevet(s)	Date de publication
EP-A-186038	02-07-86	JP-C- 1825638	28-02-94
		JP-B- 5033416	19-05-93
		JP-A- 61139873	27-06-86
		CA-A- 1245764	29-11-88
		FR-A- 2574963	20-06-86
		US-A- 4710613	01-12-87